

PERSONAL SOFTWARE

ANNO 5 N. 36
FEBBRAIO 1986
L. 4.500

LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER

Spedizione in abbonamento postale gruppo III/70



GRUPPO EDITORIALE
JACKSON
DIVISIONE PERIODICI

**SKY
VIDEOGAME
PER SHARP**

M20 PAINT

PRO.DOTTO PER APPLE

INPUT SIMULATO PER MSX

GRAFICA CON SINCLAIR QL

LIFE PER SINCLAIR QL

IL CAMPANARO PER APPLE

LUCY WRITER PER SPECTRUM

SABELLOTTI



Un breve cenno su Life per quei lettori che non ne hanno mai sentito parlare: si tratta di un gioco di simulazione inventato quasi trent'anni fa dal matematico americano Conway in cui si assiste all'evolversi nel tempo di una alquanto improbabile colonia di cellule. Questa colonia è improbabile perché si estende esclusivamente nel piano (non conosce la terza dimensione) e soprattutto perché le sue cellule nascono e muoiono seguendo delle rigide regole matematiche; nonostante ciò, Life mantiene un fascino tutto particolare, soprattutto per l'imprevedibilità delle configurazioni che possono venirsi a formare, anche partendo da una colonia di poche cellule.

Le regole che determinano la vita della colonia sono davvero molto semplici: ogni locazione nel piano contiene una cellula, che può essere viva oppure morta (oppure potete dire che contiene una cellula o non contiene niente, se non volete assolutamente ammettere che le cellule possano risorgere); una cellula morta torna a essere viva se confina con esattamente tre cellule vive (si considerano adiacenti le otto locazioni che circondano quella posizione), mentre una cellula viva può continuare a vivere se ha due oppure tre cellule vicine vive; se invece ha meno di due oppure più di tre vicini vivi essa muore (per isolamento nel primo caso e per soffocamento nel secondo).

Per non porre dei limiti alla crescita della colonia, Life dovrebbe svolgersi su un piano infinito, cosa chiaramente impossibile adoperando il computer; spesso in casi come questo si ricorre all'artificio di considerare adiacenti il bordo inferiore e quello superiore, come pure i due bordi laterali (il risultato è quello che si avrebbe giocando su una superficie toroidale); il piano di gioco rimane finito, ma in questo modo ogni cellula continua ad avere sempre otto vicini e si evitano fastidiose distorsioni ai limiti del campo.

Anche questo programma di Life per il QL adotta questo artificio per espandere logicamente il campo, che misura 24 righe per 30 colonne; è scritto esclusivamente in SuperBasic, ma è lo stesso molto veloce (una decina di secondi per colonie di una trentina di elementi), grazie a un algoritmo ottimizzato e a qual-

Anche per il QL, uno dei più famosi giochi di simulazione

di Francesco Balena

che trucchetto che spiegheremo più avanti.

Dando il Run al programma ci si porta in fase di creazione della colonia: possiamo spostarci sul piano con le frecce e lasciar cadere delle cellule vive con la barra spaziatrice, che può anche essere usata per correggere gli errori. Come spiega anche il messaggio nella finestra in basso a destra, in questa fase sono attivi altri due tasti: F1 che pulisce l'intero schermo (in pratica effettua un Run) e F2 che termina la fase di creazione e che potremo usare solo quando saremo soddisfatti della configurazione iniziale per passare alla fase di elaborazione vera e propria.

A questo punto, dopo che è stata calcolata e visualizzata la prima generazione di cellule, abbiamo la possibilità di visualizzare le generazioni successive premendo un tasto qualsiasi, a eccezione di F1 che ci rimanda alla fase di creazione (o meglio di editing, in questo caso) e di F2 che attiva l'auto-mode, in cui le generazioni vengono create senza attendere l'intervento dell'utente; per uscire dall'auto-mode è sufficiente premere un qualsiasi tasto.

Il programma

Scrivere un programma come Life sembra a prima vista molto facile (e in parte lo è); basta considerare una matrice bidimensionale che rappresenta le cellule della colonia e per ogni generazione calcolare il numero di vicini vivi per ogni posizione e decidere così per la vita o per la morte di ogni cellula.

Eppure questo metodo non funziona correttamente. Infatti poiché si suppone che le nascite e morti avvengono nello

stesso istante per tutte le cellule, avremo bisogno di due matrici, una per tener conto della situazione attuale (su cui contare il numero dei vicini) e l'altra per memorizzare la situazione che si verrà a creare all'inizio della generazione successiva. Tra una generazione e l'altra dovremo ricordarci di caricare nella prima matrice gli elementi della seconda, con una perdita di tempo non indifferente.

Quest'ultimo sistema funziona, ma risulta un po' troppo lento, perché la matrice deve essere esaminata tutta, anche se la colonia conta una minima parte di cellule vive.

Il presente programma utilizza invece un approccio alquanto diverso. Le matrici in gioco sono ancora due: la prima memorizza la situazione della colonia (uno per le cellule vive e zero per quelle morte) mentre l'altra conta per ciascuna posizione il numero dei suoi vicini vivi e viene aggiornata a ogni nascita o morte, esattamente come la prima. A esempio, se una cellula risorge, il programma memorizzerà uno nella posizione corrispondente nella prima matrice e incrementerà di uno tutte le caselle della seconda matrice adiacenti a quella della cellula appena nata. Il vantaggio di questo sistema è evidente: nel primo metodo il calcolo dei vicini veniva ripetuto per tutte le $24 \times 30 = 720$ caselle, mentre così viene effettuato solo per quelle (poche) cellule che cambiano stato da una generazione all'altra.

In realtà il programma usa due matrici per memorizzare il numero dei vicini, esattamente come nel metodo prima descritto. In questo caso occorre usare due matrici per memorizzare la situazione della colonia: una per la situazione attuale e l'altra per la situazione che si

verrà a creare all'inizio della generazione successiva.

L'altro trucco usato dal programma è quello di impiegare non delle matrici bidimensionali, bensì delle semplici stringhe, e cioè CS per la situazione della colonia (conterrà dei caratteri zero oppure uno a seconda che la cellula corrispondente sia morta o viva), pS o nS per contare il numero dei vicini vivi per ciascuna posizione (che conterranno dei numeri da zero a otto), ovviamente tutte e tre della stessa dimensione. Perché usare delle stringhe e non delle matrici bidimensionali numeriche (che tra l'altro renderebbero il programma più leggibile)? Perché di tutte le cellule del campo a noi interessano solo quelle vive, che possono morire se non hanno due o tre vicini vivi, e quelle con esattamente tre vicini vivi, che se sono morte possono tornare a vivere. Poiché queste cellule sono in genere una piccola parte di tutte le cellule del gioco, è molto conveniente fermare la nostra attenzione su di esse soltanto, rintracciandole nelle stringhe sopra citate per mezzo della velocissima funzione di Instring (vedere riga 1990) ed evitando di

prendere in considerazione tutte le cellule morte che sicuramente rimarranno tali nella prossima generazione.

Un altro vantaggio niente affatto trascurabile (in termini di tempi di elaborazione) viene dal fatto che per trasferire il contenuto di una matrice in un'altra, occorre trasferire ogni singolo elemento, mentre nel caso di due stringhe, il tutto si risolve con una semplice e rapidissima assegnazione (riga 1840).

Principali variabili globali

nri - Numero delle righe del campo di gioco (24).

nco - Numero delle colonne del campo di gioco (30).

cell - Numero delle cellule vive.

gener - Numero di generazioni elaborate.

cS - Stringa di caratteri che rappresenta la situazione della colonia (uno per le cellule vive e zero per quelle morte).

pS - Stringa che conta il numero dei vicini per ogni posizione del campo; essa viene testata nella procedura "gene-

ra" insieme alla precedente per decidere il destino di ogni cellula.

nS - Anche questa stringa conta il numero dei vicini per ogni cellula, ma a differenza della pS, viene aggiornata man mano che le cellule cambiano stato, per tener conto della situazione che si verrà a creare all'inizio della generazione seguente.

Nota: Tutte queste tre stringhe non contano 720 elementi, come ci si potrebbe aspettare, bensì $26 \times 32 = 832$ caratteri; derivano cioè da una matrice con due righe e due colonne in più, per tener conto del fatto che i bordi opposti sono considerati adiacenti e per evitare fastidiosi controlli sugli indici.

co1, co2, co3 - coefficienti (calcolati una volta per tutte nelle righe 1370-1380) che stabiliscono la corrispondenza tra il campo di gioco (quello che vediamo sullo schermo) e le stringhe che lo rappresentano internamente. A esempio, l'elemento (r,c) della matrice corrisponde all'elemento:

$$r * co1 + c + 1$$

di una delle tre stringhe cS, pS o nS.

Listato 1 - Il programma Life.

```

1000 REMark -----
           L I F E
           Fra
ncesco Balena - 1985
-----
1010 init
1020 crea
1030 REpeat loop
1040 CLS #7: INK #7,2: FLASH #7,1
1050 PRINT #7,\\\\" ELABO RAZIONE':
      FLASH #7,0
1060 genera
1070 BEEP 800,6
1080 CLS #7: INK #7,4: PRINT #7\\
      'PREMI UN TASTO'
1090 INK #7,7: PRINT #7\\\\"F1 PER RESTA
RT'\\\\"F2 PER AUTOGEN'
1100 RANDOMISE KEYROW(0)
1110 tasto=CODE(INKEYS(-1))
1120 IF tasto=232: crea: NEXT loop
1130 IF tasto=236 THEN
1140 CLS #7: INK #7,2: FLASH #7,1

```

```

1150 PRINT #7\\\\" AUTO MODE'
1160 FLASH #7,0: INK #7,7: PRINT
      #7\\\\"PREMI UN TASTO'
1170 REPEAT cont: genera: BEEP 800,6
      : IF INKEYS<>' THEN EXIT cont
1180 END IF
1190 END REPEAT loop
1200 REMark -----
           INIZIALIZZAZIONE
-----
1210 DEFine PROCedure init
1220 LOCAL j,k
1230 MODE 8
1240 OPEN #4,scr_512x256a0x0
1250 PAPER #4,0,1,0: CLS #4
1260 OPEN #4,scr_364x242a25x10
1270 PAPER #4,0: INK #4,7: OVER #4,-1:
      CLS #4: BORDER #4,1,2
1280 OPEN #5,scr_64x24a410x34
1290 PAPER #5,0: CLS #5: BORDER #5,1,7
1300 INK #5,7: STRIP #5,2: PRINT #5,
      'GENER': STRIP #5,0
1310 OPEN #6,scr_64x24a410x75
1320 PAPER #6,0: CLS #6: BORDER #6,1,7
1330 INK #6,7: STRIP #6,2: PRINT #6,
      'CELLS': STRIP #6,0
1340 OPEN #7,scr_88x126a398x120

```

Seguito listato Life.

```
1350 PAPER #7,0:CLS #7:BORDER #7,1,7
1360 nri=24: nco=30
1370 col=nco+2: co2=nri*col
1380 co3=(nri+1)*col
1390 cell=0
1400 c$= FILL$('0',(nri+2)*col)
1410 n$=c$: p$=c$
1420 END DEFine init
1430 REMark -----
      CREAZIONE DELLA 'COLONIA' -----
1440 DEFine PROCedure crea
1450 LOCAl r,c,cur$
1460 gener=0
1470 CLS #7: OVER #4,0
1480 INK #7,4:PRINT #7,' USA I '
1490 INK #7,7:PRINT #7,'\<SPACE> PER O
N/OFF'\<F1 PER CLEAR'\<F2 PER FINIRE'
1500 r=nri DIV 2: c=nco DIV 2
1510 REPEAT sposta
1520 cur$=' ': IF c$(r*col+c+1)=1
      THEN cur$='O'
1530 AT #4,r-1,c-1: PRINT #4,cur$
1540 tasto=CODE(INKEY$(5))
1550 AT #4,r-1,c-1:PRINT #4,' '
1560 IF cur$='O' THEN AT #4,r-1,c-1
      : PRINT #4,cur$
1570 SElect ON tasto
1580 ON tasto=192
1590   c=c-1: IF c=0 THEN c=nco
1600   ON tasto=200
1610   c=c+1: IF c>nco THEN c=1
1620   ON tasto=208
1630   r=r-1: IF r=0 THEN r=nri
1640   ON tasto=216
1650   r=r+1: IF r>nri THEN r=1
1660   ON tasto=32
1670   BEEP 200,3
1680   st=1-c$(r*col+c+1)*2
1690   aggiorna r*col+c+1,st
1700   ON tasto=232
1710   RUN
1720   ON tasto=236
1730   EXIT sposta
1740   END SElect
1750 END REPEAT sposta
1760 OVER #4,-1
1770 END DEFine modifica
1780 REMark -----
      NUOVA GENERAZIONE -----
1790 DEFine PROCedure genera
1800 LOCAl r,c,st,el,succ
1810 gener=gener+1
1820 AT #5,1,0: PRINT #5,FILL$( ' ',4-
      LEN(gener)):gener;
1830 AT #6,1,0: PRINT #6,FILL$( ' ',4-
```

```
      LEN(cell)):cell;
1840 p$=n$
1850 FOR r=nco+3 TO LEN(p$) STEP col
1860   p$(r+1)=n$(r+1)+p$(r+nco+1)
1870   p$(r+nco+1)=0
1880   p$(r+nco)=p$(r+nco)+p$(r)
1890   p$(r)=0
1900 END FOR r
1910 FOR c=2 TO nco+1
1920   p$(col+c)=p$(col+c)+p$(co3+c)
1930   p$(co3+c)=0
1940   p$(co2+c)=p$(co2+c)+p$(c)
1950   p$(c)=0
1960 END FOR c
1970 el=nco+3
1980 REPEAT cerca
1990   succ=primo('1' INSTR c$(el+1 TO
      ),'3' INSTR p$(el+1 TO ))
2000   IF succ=0 THEN EXIT cerca
2010   el=el+succ
2020   IF c$(el)=1 THEN
2030     IF p$(el)<>2 AND p$(el)<>3
      THEN aggiorna el,-1
2040   ELSE
2050     IF p$(el)=3 : aggiorna el,1
2060   END IF
2070 END REPEAT cerca
2080 END DEFine genera
2090 REMark -----
      AGGIORNA I VETTORI PER
      UNA 'NASCITA' O 'MORIE' -----
2100 DEFine PROCedure aggiorna (el,st)
2110 c$(el)=c$(el)+st
2120 cell=cell+st
2130 n$(el-33)=n$(el-33)+st
2140 n$(el-32)=n$(el-32)+st
2150 n$(el-31)=n$(el-31)+st
2160 n$(el-1)=n$(el-1)+st
2170 n$(el+1)=n$(el+1)+st
2180 n$(el+31)=n$(el+31)+st
2190 n$(el+32)=n$(el+32)+st
2200 n$(el+33)=n$(el+33)+st
2210 AT #4,el DIV col-1,el MOD col-2
2220 PRINT #4,'O'
2230 END DEFine aggiorna
2240 REMark -----
      FUNZIONE 'PRIMO' -----
2250 DEFine FuNction primo(a,b)
2260 IF b=0 OR (a<>0 AND a<b) THEN
      RETURN a
2270 RETURN b
2280 END DEFine primo
```

Commento al listato

1000-1020 - Si inizializzano variabili e finestre e si permette di creare la configurazione iniziale di cellule.

1030-1110 - Ciclo principale del programma; viene visualizzata la generazione successiva e si attende la pressione di un tasto. Notare l'istruzione in riga 1100 che ha l'effetto di svuotare il buffer di tastiera.

1120-1190 - A seconda di quale tasto è stato premuto si torna alla fase di edit (F1), si attiva l'auto-mode (F2) o semplicemente si elabora la generazione seguente. Notate che non c'è modo di uscire dal ciclo principale, e il programma si può arrestare solo dando il Break o il Reset.

1200-1420 - Inizializzazione delle finestre e delle variabili usate dal programma.

1430-1500 - Procedure di creazione della colonia; la stessa procedura è richia-

mata a gioco iniziato per effettuare delle piccole variazioni. La variabile curS conterrà il simbolo con cui verrà rappresentato il cursore lampeggiante (cioè "+" oppure 0 a seconda che si trovi su una casella libera o meno).

1510-1770 - Loop di attesa; notate la Select in riga 1570 che distingue le varie azioni da compiere in risposta alla pressione di un determinato tasto. Le variabili r e c memorizzano la posizione del cursore.

1780-1830 - È la Procedura che elabora una nuova generazione di cellule, ed è quindi un po' il cuore di Life. Notate come vengono stampate giustificate a destra le informazioni nelle finestre #5 e #6.

1840-1960 - La stringa p\$ contiene il numero dei vicini vivi per ciascuna posizione. I due cicli For/End For aggiornano gli elementi di p\$ che si riferiscono alle cellule del bordo (i cui vicini si trovano quindi sul lato opposto).

1970-2080 - Ciclo in cui vengono ricer-

cate le cellule vive che devono morire e viceversa: "el" punta all'elemento in esame. Notate come in riga 1990 vengono analizzate soltanto le cellule vive (il cui elemento in c\$ è uno) e le cellule con esattamente tre vicini, facendo uso della funzione Instrng e Primo.

2090-2230 - Procedura di aggiornamento, che viene richiamata in caso di nascita o morte di una cellula: "el" punta all'elemento da aggiornare e si vale +1 o -1 (rispettivamente per una nascita e una morte). Alle righe 2130-2200 viene incrementato (o decrementato) il contatore dei vicini vivi per tutte le cellule che circondano la cellula nata (o morta).

L'istruzione in riga 2220 funziona correttamente perché nella finestra #4 è attiva l'opzione di Over -1.

2240-2280 - La funzione Primo restituisce il minore tra due positivi, oppure il maggiore se uno dei due è nullo. ■



NEL PROSSIMO NUMERO TROVERETE

IN
EDICOLA
DAL
28
FEBBRAIO

MUSIC UTILITY
PER C 64

PRO.DOTTO
PER APPLE

CRUCI MSX

GRAFICA 3D
CON IL C 16

PANICO
PER SPECTRUM

ISTOGRAMMI
CON LO SHARP

RETICOLATO
PER SPECTRUM

ARCHIVIO MICRODRIVE
PER SPECTRUM

Presentazione

Lucy writer, come facilmente intuibile, è un potente wp per lo ZX Spectrum. Sfrutta al meglio le caratteristiche del calcolatore, soprattutto per chi possiede anche i microdrive. È di facile uso, essendo interattivo.

Il programma, nonostante sia scritto in Basic, consta di alcune routine in linguaggio macchina, che consentono di aumentarne la velocità nei punti dove il Basic risulterebbe troppo lento. Il programma, in fase di stampa, giustifica a destra e a sinistra il testo con un numero a scelta di caratteri per linea, provvedendo anche a dividere in sillabe le parole interrotte a fine riga per una perfetta impaginazione.

Se per qualsiasi motivo si dovesse interrompere il programma, sarà sufficiente battere "Goto Restart" ed Enter per farlo ripartire.

Struttura del programma

Il testo è contenuto in un'unica stringa ed è mostrato sul video, a pagine di 21 righe per 32 colonne: le pagine sono 35 per un totale di 23.520 caratteri, più che sufficienti per evadere la corrispondenza o per redigere brevi dattiloscritti.

Il testo non è video-orientato, cioè quello che appare sul video non coincide con ciò che verrà stampato: questa non è una limitazione perché non ha nessun interesse sapere come lo scritto sarà stampato, infatti sarà sempre conforme ai nostri voleri.

Appena lanciato, passata la presentazione, si entra in ambiente scrittura, e appare la prima pagina; sulla prima riga compare la scritta in inverse indicante l'inizio del testo: in basso, sotto una linea di separazione, è indicato il numero della pagina su cui ci si trova, (e che è la pagina corrente). Ora potete iniziare a scrivere dal punto ove è posizionato il cursore di scrittura ("") senza curarvi di andare a capo correttamente, ma scrivendo tutto di seguito come fosse tutta un'unica riga badando a inserire lo spazio all'inizio della riga successiva quando la parola precedente terminava esattamente a fine riga.

Quando si vuole andare a capolinea si preme Enter: comparirà una nota musi-

cale; quest'ultima avvertirà il calcolatore in fase di stampa che il paragrafo è finito. Se per errore la si cancella va riscritta. Resta attivo il tasto Delete (Cap Shift + 0). Tutti i tasti hanno il repeat.

Premendo Symbol Shift + e (>=) si aziona il tabulatore che sposta il cursore di cinque colonne per default. Premendo S.S. + h (↑) si aziona il modo calcolatore: verrà richiesta una formula, il cui risultato verrà stampato sul video dove era il cursore, e inserito nel testo.

Premendo S.S. + a (Stop) si passa in ambiente comandi: il loro elenco e relative spiegazioni compare in tabella 1.

Dall'ambiente scrittura, premendo Edit (C.S. + 1), si entra in ambiente editor. Ora è comparso il cursore di edit (in inverse) ed è possibile posizionarlo altrove, usando i tasti freccia, per eseguire correzioni; per sostituire la lettera del cursore basta premere il tasto corrispondente alla nuova lettera. Per più complesse modifiche bisogna ripremere Edit: si entrerà così in ambiente alterazioni. Qui con comandi a tasto unico (elencati in tabella 2) si possono apportare le modifiche al testo.

Da questo ambiente, premendo Stop (S.S. + a), si ritorna in ambiente editor; premendolo nuovamente si ripassa in ambiente scrittura con la possibilità di far tornare il cursore dov'era originaria-

mente oppure di lasciarlo dove si trovava il cursore dell'ambiente editor.

Nelle pagine successive a quella iniziale, la prima riga conterrà l'ultima riga della pagina precedente: ciò per facilitare la stesura del testo.

Spiegazione di alcuni comandi

Change - Cambia una stringa con un'altra. La ricerca è effettuata a cominciare dal primo carattere della pagina corrente, ed è possibile per stringhe fino a 20 caratteri. La stringa individuata è segnalata dallo spostamento del cursore. Ora premendo "c" avviene la sostituzione; premendo "a" la ricerca prosegue; premendo Edit (C.S. + 1) si passa in ambiente editor.

Locate - Ricerca di una stringa. Il funzionamento è simile al Change salvo che il tasto "c" non ha alcun effetto.

Print - Stampa su carta il testo perfettamente giustificato con il C.P.L. voluto, interrompendosi quando si preme S.S. + a (Stop) o si incontra il marcatore "S*S".

Alla fine del testo è infatti necessario porre il suddetto marcatore a inizio riga, altrimenti la stampante continuerà a scrivere spazi fino all'esaurimento delle pagine. Ponendo, a inizio riga, il marcatore ":" la stringa immediatamente do-

Lucy writer per Spectrum

Un potente word processor
di facile utilizzo,
particolarmente mirato
ai possessori di microdrive

di Maurizio Rogante e Andrea Nordio

ZX SPECTRUM

Tabella 1 - Elenco dei comandi.

aid	Pone i numeri di riga.
back	Indietro di una pagina.
bot	Va all'ultima pagina.
change	Cambia una parola con un'altra, a partire dalla pagina corrente.
colour	Per la scelta dei colori di Ink e Paper.
cpl n	Setta a n i caratteri per linea della stampa (default 32).
delpage	Pulisce la pagina corrente.
edit	Passa direttamente in ambiente Edit.
forw	Va avanti di una pagina.
home	Porta il cursore scrittura in alto a sinistra.
go n	Va alla pagina n e la rende corrente.
help	Mostra l'elenco dei comandi e delle funzioni di alterazione.
inform	Informazioni generali.
locate	Trova, a partire dalla pagina corrente, una parola data.
new	Comincia tutto da capo.
print	Stampa su stampante.
ret	Torna in ambiente scrittura.
sylls	Scomposizione sillabica di una parola o frase della lingua italiana.
tab n	Setta il tabulatore perché salti ogni n colonne.
top	Va alla prima pagina.
type	Stampa su stampante senza giustificare a 32 colonne (stampa di servizio).
view	Mostra tutto il testo sul video con lo scroll, premendo un tasto qualsiasi. Premendo Stop (S.S. + a) si torna ai comandi.

Comandi per la gestione della memoria di massa

save	Salvataggio
load	Caricamento con sistemazione a partire dalla prima pagina.
append	Caricamento con sistemazione a partire da una pagina specificata.

I comandi di cui sopra, se seguiti da un asterisco si riferiscono ai microdrive.

format	Per la formattazione di una nuova cartuccia, a cui viene dato il nome "Testi".
cat n	Catalogo della cartuccia nel microdrive n.
erase	Cancellazione di un file dati nel microdrive.

N.B. - Tutti i comandi in cui compare la lettera n (tipo go n) vogliono un argomento, che dev'essere intero positivo e distare almeno uno spazio dalla parola precedente.

Tabella 2 - Elenco funzioni di alterazione.

a	Aggiungere uno spazio del cursore spostando solo la linea corrente.
c	Cancella la linea corrente dal cursore in poi.
d	Cancella uno o più caratteri dopo il cursore spostando tutto il testo.
D	Cancella un carattere dopo il cursore spostando solo la linea corrente (è il duale di "a").
e	Cancella una o più righe dalla corrente verso il basso spostando in su tutto il testo.
I	Aggiunge una o più linee sopra la linea corrente, spostando in giù tutto il testo. Le ultime linee dell'ultima pagina vanno così perse.
i	Inserisce una stringa di caratteri prima del cursore.
m	Muove blocchi di linee anche da una pagina all'altra.
n	Sposta a capo la linea corrente dal cursore alla fine.
N	Sposta la linea a capo della corrente dopo il cursore. Eventuali caratteri in eccesso saranno perduti.
Stop	Torna in ambiente Edit (S.S. + a).

N.B. - I comandi Change, Locate, Print e la funzione "d" sono spiegati più diffusamente nel testo. I comandi e le funzioni vanno battuti in minuscolo. Le funzioni in maiuscolo vanno date premendo anche Caps Shift.

Figura 1 - Hardcopy del programma.

Si racconta che il principe di Conde dormì profondamente la notte avanti la giornata di Rocroi: ma, in primo luogo, era molto affaticato; secondariamente aveva già date tutte le disposizioni necessarie, e stabilito ciò che dovesse fare, la mattina.

da "I Promessi Sposi" Cap. II

Si racconta che il principe di Conde dormì profondamente la notte avanti la giornata di Rocroi: ma, in primo luogo, era molto affaticato; secondariamente aveva già date tutte le disposizioni necessarie, e stabilito ciò che dovesse fare, la mattina.

da "I Promessi Sposi" Cap. II

po di esso verrà centrata nella riga; se l'ultimo carattere è la nota, quest'ultima verrà ignorata.

L'opzione definita dal ":" ha effetto solo se utilizzata all'inizio di un capoverso.

Funzione "d" - Delete. Premendo il tasto "d" in ambiente alterazioni, si aziona la funzione delete che cancella dal testo gruppi di caratteri. Appare un cursore lampeggiante dov'era il cursore di editor, spostabile coi tasti freccia (senza C.S.). Premendo Stop si rinuncia alla funzione. I caratteri cancellati saranno quelli in inverse più il cursore. Se si tenta di cancellare oltre la fine del paragrafo, i caratteri in eccesso saranno ignorati. Non si può oltrepassare la pagina corrente. Premere Enter per far avviare la modifica.

Modifiche al programma

La subroutine che effettua materialmente la stampa su carta si trova in linea 3500, ed è predisposta per una stampante Spectrum dedicata. Per renderla compatibile alla vostra stampante (non Spectrum-dedicata) dovrete modificarla opportunamente. Ricordatevi di mettere il Return al termine della subroutine.

Chi volesse aggiungere nuovi comandi o funzioni di alterazione, sappia che la routine e la tabella dei primi inizia in linea 400, mentre la routine e la tabella delle seconde inizia a 1100.

Principali routine

- 500** - Preparazione alla giustificazione.
- 600** - Ricerca dei paragrafi nel testo per la stampa.
- 700** - Stampa di servizio (comando type).
- 800** - Scomposizione sillabica.
- 1142** - Cancellazione dei caratteri dal testo.
- 2000** - Presentazione iniziale e sistemazione del linguaggio macchina.
- 2034** - Inserimento di stringhe nel testo.
- 3000** - Giustificazione.
- 8000** - Creazione delle costanti e dell'area testo.
- 8530** - Passaggio parametri al linguaggio macchina.

Principali variabili

- R,C** - Coordinate cursore scrittura.
- R1,C1** - Coordinate cursore Edit.

TAB - Parametro di tabulazione (default cinque).

CPL - Caratteri per linea (default 32).

CC - Spazi da inserire per la giustificazione.

AA,BB - Primo e ultimo carattere nel testo per operazioni varie.

POS - Posizione nel testo del cursore di Edit.

S(20) - Array contenente il numero d'ordine dell'ultima lettera di ogni sillaba della parola esaminata. S(1) è sempre zero. Massimo 19 sillabe.

RS - Contiene il comando o la stringa da manipolare nel corso della stampa.

FS - Contiene il capoverso da trattare.

NS - Nome del testo.

TS(23520) - Il testo.

Routine in linguaggio macchina

65447 - Trova una stringa nel testo, memorizzata da 65427, e lunga al massimo 20 caratteri.

65489 - Trova la prima nota dal cursore.

65504 - Pulisce tutto il testo.

65524 - Sposta su o giù parte del testo.

Listato 1 - Il programma Lucy writer.

```

10 REM by M.Rogante & A.Nordio
11 REM
12 GO SUB 8e3: GO SUB 2e3: PAU
SE al: BORDER al: INK 4: POKE 23
624,4: CLS: GO SUB pg: LET c=al
15 LET r=si
20 PRINT AT r,c:" "
25 PAUSE al: LET k=CODE INKEYS
BEEP .01,45
35 IF k<32 OR k>164 OR k=94 TH
EN GO TO 85
40 PRINT AT r,c:CHR$ k: LET ts
(pc+r*80+c+si)=CHR$ k
45 LET c=c+1: IF c>31 THEN GO
TO 60
46 GO TO 20
60 LET c=c-80: LET r=r+1
65 IF r<22 THEN GO TO 20
75 GO TO 1000
85 IF k=13 THEN GO TO 130
90 IF k=12 THEN GO TO 145
100 IF k=7 THEN GO TO 245
105 IF k=94 THEN GO TO 180
110 IF k=226 THEN GO TO 400
115 IF k=200 THEN GO TO 1410
120 GO TO 25
130 PRINT AT r,c:"j"
135 LET ts(pc+r*80+c+si)="j": L
ET c=80: GO TO 50

```

```

147 PRINT AT r,c:" "
150 LET c=c-si
155 IF c>al THEN GO TO 175
160 LET c=31: LET r=r-si
165 IF r>=si THEN GO TO 175
170 LET r=si: LET c=al: GO TO 2
0
175 LET ts(pc+r*80+c+si)=" ": G
O TO 20
180 INPUT "Formula? ": LINE as:
GO SUB pg: IF as="" THEN GO TO
25
190 LET as=STR$ VAL as
200 IF r=21 AND c+LEN as>8031 A
ND pa=mxpg THEN GO TO 1025
210 LET ts(pc+r*80+c+si TO pc+r
*80+c+LEN as)=as
215 GO SUB mp: LET c=c+LEN as:
GO TO 45
245 PRINT INVERSE si; AT r,c:" "
250 LET r1=r: LET c1=c
260 IF INKEY$("<") THEN GO TO 26
0
265 LET c=ts(pc+r1*80+c1+si)
275 PRINT AT r1,c1: INVERSE si;
c: PAUSE al
280 LET k=CODE INKEYS
285 IF k<32 OR k>164 THEN GO TO
300
290 PRINT AT r1,c1:CHR$ k: LET
ts(pc+r1*80+c1+si)=CHR$ k: GO TO
265

```


Seguilo listato Lucy writer.

```

300 PRINT AT r1,c1:c8
301 IF k=226 THEN GO TO 360
305 IF k=7 THEN GO TO 1100
310 LET r1=r1+(k=10)-(k=11): LE
T c1=c1-(k=8)+(k=9)
312 IF r1>21 THEN LET r1=s1
314 IF r1<s1 THEN LET r1=21
316 IF c1>31 THEN LET c1=a1
318 IF c1<a1 THEN LET c1=31
320 GO TO 265
360 PRINT #s1;AT s1,a1;"OUI ?
": PAUSE a1
365 IF INKEYS="n" THEN GO TO 37
2
370 LET r=r1: LET c=c1
372 GO SUB pg: IF r>21 THEN GO
TO 1025
374 GO TO 20
400 PRINT AT r,c:ts(pc+r*mo+c+s
i)
402 GO SUB pg: INPUT FLASH s1:"
COMANDI"; FLASH a1:" ***"; LIN
E r s
404 GO SUB pg: LET k=a1: LET ar
g=-s1: IF LEN r<3 THEN GO TO 46
0
406 FOR i=s1 TO LEN r: IF r$(i
)="" THEN LET k=i: GO TO 410
408 NEXT i
410 IF k>a1 THEN LET arg=VAL r$
(k+s1 TO ): LET r$(k TO k-s1)
418 IF r$="form" THEN GO TO 105
0
420 IF r$="back" THEN GO TO 105
6
422 IF r$="save" OR r$="save:"
THEN GO TO 5000
424 IF r$="load" OR r$="load:"
THEN GO TO 5006
426 IF r$="inform" THEN GO TO 7
070
428 IF r$="edit" THEN GO TO 102
0
429 IF r$="colour" THEN GO TO 5
050
430 IF r$="ret" THEN GO TO 20
432 IF r$="go" THEN GO TO 4100
434 IF r$="top" THEN GO TO 950
436 IF r$="bot" THEN GO TO 956
438 IF r$="delpage" THEN GO TO
4110
440 IF r$="view" THEN GO TO 145
0
442 IF r$="tab" THEN GO TO 1420
444 IF r$="change" THEN GO TO 4
205
446 IF r$="locate" THEN GO TO 4
200
448 IF r$="sylls" THEN GO TO 15
10
450 IF r$="home" THEN GO TO 150
0
451 IF r$="cat" THEN GO TO 5034
452 IF r$="append" OR r$="appen
d:" THEN GO TO 5012
453 IF r$="format" THEN GO TO 5
032
454 IF r$="new" THEN RUN
456 IF r$="print" OR r$="type"
THEN GO TO 600
458 IF r$="cpl" THEN GO TO 720
460 IF r$="aid" THEN GO TO 690
462 IF r$="erase" THEN GO TO 50
38
464 IF r$="help" THEN GO TO 120
0
466 PRINT #s1;AT s1,a1;us;AT s1
,a1;"NON ESISTE *****"; BEE
P 1,0: PAUSE 60: GO TO cmd
468 PRINT #s1;AT s1,a1;"ARGOMEN
TO ERRATO": BEEP 1,0: PAUSE 60:
GO TO cmd
500 LET i=s1: IF i>brk THEN LE
T ps=fs: GO SUB pt: RETURN
505 LET as=fs: TO i+cpl-s1)
510 LET rs=as: TO cpl): LET s$=
as(cpl+s1 TO ): LET j=i+cpl
512 IF INKEYS=" STOP " THEN RET
URN

```

ZX SPECTRUM

```

514 LET bs=s$+fs(j TO j+cpl-s1)
518 IF r$(cpl)="" OR bs(s1)=""
" THEN GO TO 552
520 FOR k=cpl-s1 TO s1 STEP -s1
: IF r$(k)<>" " THEN NEXT k
525 FOR w=pr TO cpl: IF bs(w)<>
" " THEN NEXT w
530 LET js=r$(k+s1 TO )+bs( TO
w-s1): GO SUB 800
535 FOR v=s1 TO 20: IF s(v)<=cp
l-s1-k THEN NEXT v
540 IF s(v-s1)=a1 THEN GO TO 55
0
545 LET rs(k+s1 TO )=js( TO s(v
-s1))+"-": LET bs=js(s(y-s1)+s1
TO )+bs(v TO ): GO TO 552
550 LET r$(k+s1 TO )=s$: LET bs
=j$+bs(v TO )
552 GO SUB 3e3: LET i=j: IF i>=
brk THEN GO TO 580
554 LET as=bs: IF LEN as<cpl TH
EN LET as=as+es( TO cpl-LEN as)
556 GO TO 510
560 IF LEN bs>cpl THEN GO TO 59
0
585 LET rs=bs: GO SUB 3078: RET
URN
590 LET rs=bs( TO cpl): LET bs=
bs(cpl+s1 TO )
592 IF bs=es( TO LEN bs) THEN G
O SUB 3078: RETURN
594 GO TO 518
600 GO SUB 9s: IF r$="type" THE
N GO TO 700
604 GO TO 608+22*(ts(aa)=""")+4
46*(ts(aa TO aa+pr)=""s+s")
610 GO SUB 1600
616 LET fs=ts(aa TO i+j-pr): LE
T r$=fs+es( TO cpl-LEN fs+cpl(I
NT (LEN fs/cpl)))
618 LET brk=LEN fs-cpl+s1: LET
aa=i+no: IF aa>bb THEN GO TO 105
4
620 GO SUB 500: IF INKEYS=" STO
P " THEN GO TO restart
622 GO TO 604
630 LET rs=ts(aa+s1 TO aa+no31)
+" ": LET aa=aa+no
632 FOR i=cpl TO pr STEP -s1: I
F r$(i)<>" " THEN GO TO 636
634 NEXT i: GO TO 604
636 IF r$(i)="" THEN LET i=i-s
1
638 LET rs=us( TO (cpl-LEN r$(
TO i))/pr)+r$( TO i): LET ps=rs:
GO SUB pt: GO TO 604
690 GO SUB 2100: PAUSE a1: GO T
O 1054
700 FOR i=aa TO bb STEP no: LET
ps=ts(i TO i+no31): GO SUB pt
702 IF INKEYS=" STOP " OR r$( T
O 3)=""s+s" THEN GO TO restart
704 NEXT i: GO TO restart
720 IF arg<1 THEN GO TO 468
722 LET cpl=arg: DIM es(cpl+s1)
: GO SUB pg: GO TO cmd
750 LET ts(pos-c1 TO pos-s1)+""+ts(p
os TO pc+r1*mo+no31): GO TO 1100
754 LET ts(pos TO pos-c1+no31)=
ts(pos+s1 TO pos-c1+no31): GO TO
1180
770 INPUT "Quante linee? ";l: L
ET fa=k$="l")
774 IF fa THEN GO SUB 1187: GO
TO 1160
776 GO SUB 1154: GO TO 1180
800 LET ks=js: DIM s(20): LET p
p=pr: LET ls=LEN js
805 IF ks(LEN ks)<CHR$ no OR ks
(LEN ks)>CHR$ 164 THEN LET ks=ks
( TO LEN ks-s1): GO TO 805
810 LET h=LEN ks
812 IF h<=2 THEN GO TO 872
814 LET li=s1
816 GO SUB 376

```

Seguito listato Lucy writer.

```
818 IF ll=0 THEN GO TO 856
820 LET ll=ll
822 LET ii=ii+si
824 IF ii>h THEN GO TO 872
826 IF ks(ii)="h" OR ks(ii)="H"
THEN GO TO 822
828 GO SUB 876
830 IF ll=al THEN GO TO 848
832 IF ll=2 OR ll=2 THEN GO TO
840
834 LET jj=ii-si
836 GO SUB 896
838 GO TO 810
840 LET ii=ii+1
842 IF ii>h THEN GO TO 872
844 LET ll=ll
846 GO TO 828
848 LET ii=ii+1
850 IF ii>h THEN GO TO 872
852 GO SUB 876
854 IF ll=al THEN GO TO 860
856 LET jj=ii-pr
858 GO TO 836
860 GO SUB 804
862 IF s=0 THEN GO TO 834
864 GO TO 856
866 LET ii=ii+1
868 IF ii>h THEN GO TO 872
870 GO TO 816
872 LET s(pp)=ls
874 RETURN
876 LET (s=ks(ii)
880 IF (s="i" OR (s="I" OR (s="
" OR (s="U" THEN GO TO 892
882 IF (s="a" OR (s="A" OR (s="
" OR (s="E" OR (s="o" OR (s="O"
THEN GO TO 886
884 LET (l=al
886 RETURN
888 LET (l=si
890 RETURN
892 LET (l=pr
894 RETURN
896 LET s(pp)=jj+s(pp-si)
900 LET pp=pp+si: LET ks=ks(jj+
si TO )
902 RETURN
904 LET (s=ks(ii-si)
906 LET (s=ks(ii)
910 IF (s=ss THEN GO TO 924
912 IF (s="g" AND ss="n" THEN G
O TO 920
914 IF (s="l" OR (s="m" OR (s="
n" OR (s="r" OR (s="v" OR (s="z"
THEN GO TO 924
918 IF (s="s" OR ss="r" THEN GO
TO 928
920 IF (s="t" THEN GO TO 924
922 IF ss="h" OR ss="l" THEN GO
TO 928
924 LET s=al
926 RETURN
928 LET s=si
930 RETURN
932 LET pa=si: LET pc=-no: GO T
O 1054
934 LET pa=bxpg: LET pc=pa+nc-n
c-no: GO TO 1054
1030 IF pa=bxpg THEN GO TO 1025
1005 CLS: LET pa=pa+si: LET pc=
pc+nc
1010 PRINT AT al,al;ts(pc+si TO
pc+no): AT si,al;ts(pc+no33 TO pc
+nc+no): GO SUB pg: GO TO 15
1020 LET r=21: LET r1=r: LET c1=
al: GO TO 265
1025 PRINT (si: AT si,al;w$; AT si
al; "NON C'E' PIU' SPAZIO!": SEE
P si,al: PAUSE 60: GO SUB pg: GO
TO 1020
1050 IF pa=bxpg THEN GO TO cmd
```

```
1052 LET pa=pa+si: LET pc=pc+nc
1054 GO SUB ap: GO TO cmd
1056 IF pa=si THEN GO TO cmd
1058 LET pa=pa-si: LET pc=pc-nc:
GO TO 1054
1100 PRINT AT r1,c1: INVERSE si;
cs; (si: AT si,al;w$; AT si,al; "MOD
O ALTERAZIONI": LET pos=pc+r1+no
+c1+si
1102 PAUSE al: LET ks=INKEY$
1104 IF ks="c" THEN GO TO 1130
1106 IF ks="i" THEN GO TO 2020
1108 IF ks="l" THEN GO TO 770
1110 IF ks="e" THEN GO TO 770
1112 IF ks="a" THEN GO TO 2200
1114 IF ks="d" THEN GO TO 2070
1116 IF ks="d" THEN GO TO 754
1118 IF ks="n" THEN GO TO 2250
1120 IF ks="a" THEN GO TO 750
1122 IF ks=" " STOP THEN GO TO 1
140
1114 IF ks="N" THEN GO TO 2253
1120 GO TO 1102
1130 LET ts(pos TO pc+r1+no+no)=
es: GO TO 1180
1140 PRINT AT r1,c1;ts(pos): GO
SUB pg: GO TO 265
1142 LET aa=pos-c1: LET bb=bxca:
GO SUB 1600: LET a=6+PEEK 23627
+256+PEEK 23628: POKe 65534,176:
IF i+j-pos(n THEN LET n=i+j-pos
1143 IF i+j)bxca THEN LET v1=pos
+n+a-si: LET v2=pos+a-si: LET v3
=bxca-v1+a: GO TO 1182
1144 LET (s=ts(pos+n TO i+no31):
LET (s(i+no31-n+si TO i+no31)=w
s: LET (s(pos TO i+no31-n)=f$
1145 IF j)n THEN RETURN
1146 LET k=INT ((ABS (j-n)+no)/n
0): LET v1=i+no+a-si: LET v2=i+m
o-no+i+a-si: LET v3=bxca-i-no
1148 GO SUB 8530: RANDOMIZE USR
65524
1150 LET (s(bxca-no+k TO )=w$: R
ETURN
1152 IF v3>al THEN GO SUB 8530:
RANDOMIZE USR 65524: LET (s(bxca
-n+si TO )=w$: RETURN
1153 LET (s(pos TO )=w$: RETURN
1154 IF bxca-(pc+r1+si+no)(<no
THEN RETURN
1156 LET pos1=pos: LET pos=pos-c
1: LET c2=c1: LET c1=al: LET n=n
0+l: GO SUB 1142: LET pos=pos1:
LET c1=c2: RETURN
1160 GO SUB pg: IF pa=si THEN PR
INT AT al,al;ts(pc+si TO pc+no)
1162 PRINT AT si,al;ts(pc+no33 T
O pc+nc+no): RETURN
1180 GO SUB ap
1182 LET cs=ts(pos): GO TO 1100
1188 IF (no)bxca-pos+c1 THEN LE
T (s(pos-c1 TO )=w$: RETURN
1190 LET pos1=pos: LET pos=pos-c
1: LET c2=c1: LET c1=al: LET as=
": FOR z=si TO l: LET as=as+w$
NEXT z: GO SUB 2034: LET pos=po
s1: LET c1=c2: RETURN
1200 CLS: PRINT INVERSE si; FLA
SH si; "MOD COMANDI"
1210 PRINT "for", "back", "infor
m", "edit", "ret", "go n", "top", "bo
t", "delpage", "view", "tab", "chang
e", "locate", "sylls", "home", "neu",
"print", "type", "cp(", "aid", "hel
p"
1215 PRINT "save", "save+", "load",
"load+", "append", "append+", "for
mat", "cat", "erase"
1220 PRINT INVERSE si; FLASH s
i; "MOD ALTERAZIONI"
1230 PRINT "c , i , l , e
, b , d O , n , N , a
, STOP"
1306 PAUSE al: GO TO 1054
1410 PRINT AT r1,c1;ts(pc+r1+no+c+s
i): LET c=tab+(INT (c/tab)+1)
1412 IF c>31 THEN LET c=al
1414 GO TO 20
1420 IF arg)no31 OR arg<si THEN
GO TO 468
1422 LET tab=arg: GO SUB pg: GO
TO cmd
```

Seguito listalo Lucy writer.

```

1450 CLS : PRINT AT 21,a1; POKE
23592,255
1455 FOR i=s1 TO mxca STEP a0
1460 PRINT ts(i TO i+m031)
1462 IF INKEYS="" THEN GO TO 146
2
1464 IF INKEYS="" STOP " THEN GO
TO restart
1465 NEXT i
1470 PRINT #s1;"Premi un tasto":
PAUSE a1: GO TO restart
1500 LET r=s1: LET c=a1: GO TO c
ad
1510 CLS : PRINT INVERSE 1;"
SCOMPOSIZIONE SILLABICA " : P
LOT a1,166: DRAU 255,a1: PLOT a1
,164: DRAU 255,a1
1512 INPUT "PAROLA O FRASE DA SI
LLABARE ",,js: PRINT AT 3,a1;"S
OLUZIONE:"
1514 GO SUB 600
1520 FOR i=pr TO 20
1525 IF s(i)<>LEN js THEN PRINT
js(s(i-s1)+s1 TO s(i)); "-" : NEX
T i
1530 PRINT js(s(i-s1)+s1 TO ) : P
AUSE a1: GO TO 1054
1600 LET dd=aa+5: POKE 65494,INT
(dd/256): POKE 65493,dd-256+PEE
K 65494
1602 LET v1=s1+bb-aa: POKE 65498
,INT (v1/256): POKE 65497,v1-256
+PEEK 65498
1604 LET v2=aa+v1-USR 65489-pr:
LET i=(INT (v2/m0)+m0)+s1: LET j
=v2-i+pr
1606 IF i>mxca THEN LET i=mxca-m
031: LET j=m0: RETURN
1608 RETURN
2002 FOR i=65447 TO 65535: READ
a$: POKE i,VAL a$: NEXT i
2010 BORDER 5: PAPER a1: CLS : I
NK 5: PLOT 56,72: DRAU OVER s1,9
0,90,243: INK 3: FLASH s1: PRINT
BRIGHT s1,AT 9,14;"LUCY": AT 11,
13: INVERSE s1;"WRITER": FLASH a
1
2012 RETURN
2020 INPUT a$: IF a$="" STOP " OR
LEN a$>mxca-pos THEN GO TO 1180
2022 GO SUB 2034: GO TO 1180
2034 LET aa=pos-c1: LET bb=mxca
2036 LET a=5+PEEK 23627+256*PEEK
23628: POKE 65534,184: GO SUB 1
600: IF i+j+LEN a$>mxca THEN GO
TO 2050
2038 LET k=INT ((LEN a$+j)-s1)/m0
: IF LEN a$<m0-j THEN GO TO 204
8
2040 LET v1=mxca-m0+k+a: LET v2=
mxca+a: LET v3=v1-a-i-m0+s1
2046 IF v3>al THEN GO SUB 8530:
RANDOMIZE USR 65524
2048 LET ts=ts(pos TO i+j-s1): L
ET ts(pos TO i+m0+k+m031)=w$: LE
T ts(pos TO pos+LEN a$)=a$:
2050 LET ts(pos+LEN a$ TO pos+LE
N a$+LEN ts)=ts: RETURN
2060 LET v2=mxca+a: LET v1=v2-LE
N a$: LET v3=s1+v1-a-pos
2062 GO SUB 8530: RANDOMIZE USR
65524
2064 LET ts(pos TO pos+LEN a$-s1
)=a$: RETURN
2070 PRINT #s1,AT s1,a1;w$: AT s1
,a1;"DELETING": LET r2=r1: LET c
2=c1: LET pos1=pos
2072 PRINT AT r2,c2: FLASH NOT C
ODE INKEYS: INVERSE s1,ts(pos1):
LET k=CODE INKEYS
2074 IF k=13 THEN LET n=pos1-pos
+s1: GO SUB 1142: GO TO 1180
2076 IF k=226 THEN GO TO 1180
2078 IF k=55 AND c2<m031 THEN PR
INT AT r2,c2: INVERSE s1,ts(pos1
): LET c2=c2+s1: LET pos1=pos1+s
1: GO TO 2072
2080 IF k=53 AND c2>a1 AND NOT (
r2=r1 AND c2<=c1) THEN PRINT AT
r2,c2,ts(pos1): LET c2=c2-s1: LE
T pos1=pos1-s1: GO TO 2072
2082 IF k=54 AND r2<21 THEN PRIN

```

ZX SPECTRUM

```

T AT r2,c2: INVERSE s1,ts(pos1 T
O pos1+m0): LET r2=r2+s1: LET po
s1=pos1+m0: GO TO 2072
2084 IF k=55 AND r2>r1 AND NOT (
r2=r1+s1 AND c2<=c1) THEN LET r2=
r2-s1: LET pos1=pos1-m0: PRINT A
T r2,c2,ts(pos1 TO pos1+m0)
2086 GO TO 2072
2100 FOR i=s1 TO 21: PRINT AT i,
a1: INVERSE s1;" " AND i<10;i, I
NVERSE a1;" ";ts(pc+i+m0+s1 TO P
c+i+m0+m031-pr)
2102 NEXT i: RETURN
2200 PRINT #s1,AT s1,a1;"Inserim
ento parametri": GO SUB 2100
2205 INPUT "Da quale pagina? ":P
1: INPUT "A quale pagina? ":P2
2210 INPUT "Da quale riga? ":l1:
INPUT "A quale riga? ":q1: INPU
T "In quale riga? ":l2: LET q1=q
1-(l1+s1
2220 IF p1>mxpg OR p1<=1 OR (l1<
1 OR p2>mxpg OR (l2<=1 OR p2<=1 O
R (l1>21 OR q1<=1 OR q1>21 OR l2>
21 OR 22-(l2-q1) THEN PRINT #s1,AT
s1,a1;"PARAMETRI ERRATI": BEEP
s1,70: PAUSE 60: GO TO 2205
2225 LET ts(p2+nc-nc+s1+(l2-s1)+
m0 TO p2+nc-nc+(l2-s1)+m0+q1+m0)
=ts(p1+nc-nc+s1+(l1-s1)+m0 TO p1
+nc-nc+(l1-s1)+m0+q1+m0): GO TO
1180
2250 LET ts(pc+(r1+s1)+m0+s1 TO
pc+(r1+s1)+m0+m0)=ts(pos TO pc+r
1+m0+m0)
2252 LET ts(pos TO pc+r1+m0+m0)=
a$: GO TO 1180
2258 IF axpg=pa AND r1=22 THEN G
O TO 1180
2260 LET ts(pos TO pc+r1+m0+m0)=
ts(pc+(r1+s1)+m0+s1 TO pc+(r1+s1
)+m0+m0): LET ts(pc+(r1+s1)+m0+s
1 TO pc+(r1+s1)+m0+m0)=w$: GO TO
1180
3000 FOR k=pr TO s1 STEP -s1
3015 IF rs(k)="" THEN NEXT k: G
O TO 3064
3025 LET cc=cp(-k): LET cc1=cc: L
ET rs=rs( TO k)
3030 FOR n=s1 TO LEN rs: IF rs(n
)="" THEN NEXT n
3032 IF n=pr THEN LET rs=rs(pr T
O ): LET cc=cc+s1: LET cc1=cc
3034 IF cc=al THEN GO TO 3054
3036 FOR w=n TO LEN rs
3037 IF rs(w)="" THEN GO TO 304
8
3038 NEXT w
3040 IF cc=cc1 THEN GO TO 3064
3042 IF cc=al THEN GO TO 3036
3044 GO TO 3064
3046 LET rs=rs( TO w)+" "+rs(w+s
1 TO ): LET cc=cc-s1: IF cc=al T
HEN GO TO 3064
3050 FOR l=w TO LEN rs: IF rs(l)
()="" THEN LET w=l: GO TO 3037
3052 NEXT l: GO TO 3036
3064 LET ps=rs: GO SUB pt: RETUR
N
3078 IF rs(s1)<>" " THEN GO TO 3
084
3080 LET rs=rs(pr TO )+" ": GO T
O 3078
3499 REM
3500 REM ** ROUTINE DI STAMPA **
3501 REM
3502 LPRINT ps: RETURN
4100 IF arg>mxpg OR arg<=1 THEN
GO TO 460
4102 LET pa=arg: LET pc=pa+nc-nc
-m0: GO TO 1054
4110 LET ts(pc+m033 TO pc+nc+m0)
=a$: GO SUB sp: GO TO 1500
4200 LET cf=a1: INPUT "COSA TROV
07 ":cs: GO TO 4208

```

ZX SPECTRUM

Seguito listato Lucy writer.

```

4205 LET cf=si: INPUT "VECCHIO "
:cs;"NUOVO ";bs
4206 GO SUB pg: IF LEN cs>20 THE
N PRINT #si;AT si,al;us;AT si,al
;"STRINGA TROPPO LUNGA (max 20)"
: PAUSE 60: GO TO cmd
4208 FOR f=si TO LEN cs: POKE f+
65426,CODE cs(f): NEXT f
4210 LET jj=LEN cs: POKE 65457,j
j-si: LET r1=si: LET c1=-si
4212 LET z1=pc+r1*mo+c1+7: POKE
65452,INT (z1/256): POKE 65451,z
1-256+PEEK 65452: LET z2=mxca-jj
-z1+6: POKE 65456,INT (z2/256):
POKE 65455,z2-256+PEEK 65456
4214 LET z=z1+z2-USR 65447-6: IF
z<=mxca-jj-si THEN GO TO 4220
4216 PRINT #si;AT si,al;us;AT si
,al;"STRINGA NON TROVATA.": PAUS
E 60: GO TO 1054
4220 LET pa=si+INT (z/nc): LET p
c=pa+nc-nc-mo: LET k=z-nc+INT (z
/nc): LET r1=INT ((k-si)/mo)+si:
LET c1=k-mo+(r1-si)-si
4222 GO SUB pg: GO SUB ap
4224 PRINT AT r1,c1: INVERSE si;
ts(z)
4226 IF CODE INKEY$=7 THEN GO TO
4226
4228 IF INKEY$="a" THEN GO TO 42
12
4230 IF NOT (INKEY$="c" AND c1)
THEN GO TO 4226
4232 LET pos=pc+r1*mo+c1+si: IF
LEN bs<jj THEN LET n=jj-LEN bs:
GO SUB 1142: GO TO 4236
4234 LET as=us( TO LEN bs-jj): G
O SUB 2034
4236 LET ts(pc+r1*mo+c1+si TO pc
+r1*mo+c1+LEN bs):bs: GO TO 4222
5000 CLS: PRINT AT 10,13; FLASH
si;"SAVING": GO SUB 5020: GO SU
B 5022
5002 IF rs="save" THEN SAVE n$CO
DE ind,qua: GO TO restart
5004 GO SUB 5026: SAVE #m";j;n$
CODE ind,qua: GO TO restart
5006 CLS: PRINT AT 10,12; FLASH
si;"LOADING": RANDOMIZE USR 655
04: GO SUB 5020: LET qua=mxca
5008 IF rs="load" OR rs="append"
THEN LOAD n$CODE ind,qua: GO TO
restart
5010 GO SUB 5026: LOAD #m";j;n$
CODE ind,qua: GO TO restart
5012 CLS: PRINT AT 10,13; FLASH
si;"APPEND": INPUT "Da quale pa
gina? ";k: IF k<si OR k>mxpg THE
N GO TO 5012
5014 GO SUB 5020: LET i=k+nc-nc+
si: LET ind=ind+i-si: LET qua=mx
ca-i: GO TO 5008
5020 INPUT "Nome del testo (max.
10) ", LINE n$: LET ind=6+PEEK 2
3627+256+PEEK 23628: RETURN
5022 POKE 65451,6: POKE 65452,al
: POKE 65455,224: POKE 65456,91:
POKE 65467,2: POKE 65427,36: PO
KE 65428,42: POKE 65429,36
5024 LET qua=mxca-USR 65447-si:
IF qua>mxca-mo31 THEN LET qua=mx
ca
5026 RETURN
5028 INPUT "Quale microdrive? ";
j: IF j<si OR j>8 THEN GO TO 502
8
5030 RETURN
5032 CLS: PRINT AT 10,13; FLASH
si;"FORMAT": GO SUB 5026: STOP
: FORMAT #m";j;"testi": GO TO 50
36

```

```

5034 LET j=arg: IF j<si OR j>8 T
HEN GO TO 465
5036 CLS: CAT J: PAUSE al: GO T
O restart
5038 CLS: PRINT AT 11,12; FLASH
si;"ERASING": GO SUB 5020: GO S
UB 5028: INPUT INVERSE si;" CONF
ERMI? "; INVERSE al;" "; LINE as
: IF as<>"si" THEN GO TO restart
5040 ERASE #m";j;n$: GO TO resta
rt
5050 INPUT "Ink? ";j;" Paper? ";
k
5052 BORDER k: INK J: PAPER k: P
OKE 23624,j+8*k: GO SUB mp: GO T
O cmd
7070 CLS: PRINT INVERSE 1;"INFO
RAZIONI"
7072 PRINT "NOME DEL TESTO :
";n$
7076 PRINT "PAGINA CORRENTE :
";pa
7078 PRINT "COORDINATE CURSORE
";r1;" - ";c
7080 PRINT "PARAMETRO DI TABULA
ZIONE ";tab
7082 PRINT "CARATTERI PER LINEA
";cpl
7084 PRINT #si;"PREMI UN TASTO P
ER RITORNARE"
7094 PAUSE 0
7096 GO TO 1054
8000 DIM ts(23520): LET mxca=LEN
ts: LET qs=8516: LET pt=3500: L
ET pg=8506: LET mp=1160: LET cmd
=402: LET al=0: LET si=1: LET pr
=2: LET mo31=31: LET mo=32: LET
mo33=33: LET nc=672
8010 DIM us(mo): LET restart=950
: LET mxpg=mxca/nc: LET pa=si: L
ET pc=-mo: LET tab=5: LET cpl=mo
: DIM es(cpl+si)
8100 FOR f=al TO 7: READ as: POK
E USR "j"+f,VAL as: NEXT f
8102 FOR f=al TO 7: READ as: POK
E USR #m"+f,VAL as: NEXT f
8104 POKE 23609,40
8200 DATA "0", "12", "10", "8", "56",
"120", "48", "0", "0", "255", "2
55", "255", "255", "0", "0",
8202 DATA "42", "75", "92", "1", "6",
"0", "9", "1", "224", "91", "17", "14",
7", "255", "26", "237", "177", "192",
"229", "197", "6", "1", "120", "254",
"0", "40", "9", "43", "19", "35", "26",
"190", "32", "5", "16", "248", "193",
"225", "201", "193", "225", "24", "2
24"
8204 DATA "42", "75", "92", "1", "6",
"0", "9", "1", "224", "91", "62", "14",
4", "237", "177", "201", "42", "75", "14",
92", "1", "6", "0", "9", "1", "224", "9
1", "62", "32", "119", "35", "11", "12",
0", "177", "200", "24", "246", "11", "33",
0", "0", "17", "0", "0", "1", "0", "0",
"237", "184", "201"
8400 RETURN
8506 PRINT #si;AT al,al;"
";us;AT
si,al;"PAGE: ";pa
8510 IF pa=mxpg THEN PRINT #si;A
T si,10; INVERSE si;" ** FINE DE
L TESTO **"
8512 IF pa=si THEN PRINT AT al,a
l; INVERSE si;" ** INIZIO DEL TE
STO **"; INVERSE al;"
8514 RETURN
8520 CLS
8522 INPUT "Dalla pag.? ";aa;" A
lla pag.? ";bb
8524 IF aa<si OR aa>mxpg OR bb<si
OR bb>mxpg OR bb<aa THEN GO TO
8522
8526 LET aa=aa+nc-nc+si: LET bb=
bb+nc: CLS: RETURN
8530 POKE 65526,INT (v1/256): PO
KE 65525,v1-256+PEEK 65526
8532 POKE 65529,INT (v2/256): PO
KE 65528,v2-256+PEEK 65529
8534 POKE 65532,INT (v3/256): PO
KE 65531,v3-256+PEEK 65532
8536 RETURN

```

Come dice il proverbio "non tutte le ciambelle riescono col buco" e anche il QL non fa eccezione a questa regola, mostrando nella grafica alcune imperfezioni. Gli unici nei sono rappresentati dal sistema di riempimento delle figure chiuse e dalla gestione delle coordinate. E di questo secondo aspetto che intendiamo occuparci in questo articolo. Per l'indirizzamento dei comandi grafici viene utilizzato un sistema di coordinate cartesiane standard indipendente dal modo grafico usato, a quattro o a otto colori, e dalla posizione del pixel. E inoltre possibile, per mezzo del comando Scale, spostare l'origine di questo sistema di riferimento relativamente alla finestra usata e modificarne anche la scala. Purtroppo però non è possibile definire indipendentemente la scala sui due assi, ma solo globalmente.

Segmenti di uguale lunghezza (in cm) sullo schermo avranno sempre anche la stessa lunghezza rispetto alle coordinate grafiche, sia che siano verticali che orizzontali.

Se da una parte ciò fa sì che i quadrati risultino sullo schermo effettivamente... quadrati, indipendentemente dal modo grafico usato e compensando la diversa densità dei pixel nelle due direzioni, d'altra parte non permette di indirizzare la grafica per pixel.

Questa carenza può risultare importante, a esempio, dovendo disegnare delle figure non troppo grandi, in quanto risulteranno lievemente diverse a seconda della zona dello schermo, o meglio a seconda del valore delle coordinate in cui verranno disegnate. Nel caso in cui l'ampiezza verticale grafica di una finestra sia stata fissata uguale a quella in pixel, tale deformazione, dovuta agli arrotondamenti eseguiti dal sistema nel far corrispondere i valori delle coordinate ai singoli punti indirizzabili, risulterà presente solo nel senso della larghezza, modificando le proporzioni delle diverse parti.

È possibile sopperire a questa mancanza per mezzo di una formula che fornisca le coordinate grafiche corrispondenti a ogni singolo punto indirizzabile dello schermo. Consideriamo per primo il caso dell'alta risoluzione: per ricavare questa formula si deve innanzitutto valutare il rapporto: tra i pixel oriz-

zontali e quelli verticali; tra il numero dei pixel che compongono una linea della stessa lunghezza, misurata nel sistema di coordinate grafiche, nelle due direzioni orizzontale e verticale. Il valore da noi trovato per questo rapporto, determinato misurando la larghezza grafica di finestre di dimensioni note, è 1,35. Se le coordinate grafiche sono state settate, con il comando Scale, in modo che l'altezza della finestra (rispetto alla grafica) sia uguale all'ampiezza verticale in pixel e l'origine sia nell'angolo inferiore sinistro della finestra (con Scale WH, 0,0; dove WH è il numero di pixel sull'asse verticale della finestra, per intenderci il secondo parametro usato con i comandi Window e Open), allora un pixel di coordinate (rispetto sempre all'angolo inferiore sinistro) M,N sarà individuato dalle coordinate grafiche:

$$X = M/1,35$$

e:

$$Y = N$$

Nel caso più generale in cui le coordinate grafiche siano state inizializzate con il comando Scale H, OrgX,OrgY si dovranno modificare le semplici formule viste sopra:

$$X = (M/Cx) + OrgX$$

e:

$$Y = (N/Cy) + OrgY$$

dove:

$$Cx = 1,35 * WH/H$$

e:

$$Cy = WH/H$$

È importante ricordare che sia OrgX che

OrgY sono dei numeri con segno.

Nel caso si lavori in bassa risoluzione, cioè con otto colori, per adattare le formule prima viste è sufficiente definire al posto di Cx, $Cx8 = Cx/2$. Poiché sia Cx (o Cx8) che Cy non dipendono dal punto da individuare: in un eventuale programma che dovesse tracciare delle linee tra due pixel, potrebbero venire definiti una sola volta. In questo programma sarebbe comodo definire due funzioni che operino la conversione da pixel a grafica, a esempio:

```
DEF FN Xg (Xp,Cx,OrgX)
LOCAL X
X = (Xp/Cx) + OrgX
RETURN X
END DEF Xg
```

e:

```
DEF FN Yg (Yp,Cy,OrgY)
LOCAL Y
Y = (Yp/Cy) + OrgY
RETURN Y
END DEF Yg
```

Un altro sistema per indirizzare per punti, particolarmente utile e veloce quando si debbano disegnare figure composte da rettangoli, consiste nell'utilizzo del comando Block che permette un indirizzamento misto, per pixel e grafico. Il difetto di questo comando è che se il rettangolo esce dai limiti della finestra viene generato un messaggio di errore e l'eventuale programma si ferma, cosa che non accade con i comandi di pura grafica. ■

Indirizzamento della grafica per pixel

Gestiamo le capacità grafiche del QL

di Massimo Modelli